

Improved Reliability-Based Iterative Decoding of LDPC Codes Based on Dynamic Threshold

Ma Zhuo and Du Shuanyi

A serial concatenated decoding algorithm with dynamic threshold is proposed for low-density parity-check codes with short and medium code lengths. The proposed approach uses a dynamic threshold to select a decoding result from belief propagation decoding and order statistic decoding, which improves the performance of the decoder at a negligible cost. Simulation results show that, under a high SNR region, the proposed concatenated decoder performs better than a serial concatenated decoder without threshold with an E_b/N_0 gain of above 0.1 dB.

Keywords: Channel coding, low-density parity-check code, reliability decoding, log-likelihood ratio accumulation.

I. Introduction

A reliability-based decoding algorithm is used to resolve performance degradation when a belief propagation (BP) algorithm is adopted to decode low-density parity-check (LDPC) codes with short and medium code lengths [1]. The performance degradation is caused by the short cycles in the Tanner graph of these codes, which cannot be ignored [2] in the case of short and medium code lengths.

References [1] and [3] perform reliability-based order statistic decoding (OSD) after every BP iteration. The difference is that in [3], a cyclic redundancy check (CRC) is added after LDPC encoding. The CRC can be used either to check the correction of the OSD or as the criterion for terminating BP iterations.

Considering the complexity of OSD decoding, another type of decoding scheme called the serial concatenated decoding is more acceptable. In this scheme, the OSD algorithm is performed only after the last BP iteration. However, the log-likelihood ratio (LLR) oscillation phenomenon observed in [4] decreases the credibility of the LLR of the last iteration as a reliability measure. This problem is resolved in [5] by accumulating the LLR over the entire number of iterations as the reliability measure. Reference [6] generalizes such an LLR accumulation to the probability domain. Reference [7] uses a conditioned LLR to feed an OSD decoder, which also leads to a serial concatenated decoding scheme. Another method of combining a BP algorithm and reliability-based decoding algorithms is reported in [8], where the reliability-based decoding is applied alternatively to BP. The relief-based decoder in [8] uses the initial soft information instead of the soft values delivered by the last BP iteration whenever BP does not converge.

Manuscript received Dec. 12, 2013; revised Apr. 1, 2015; accepted Apr. 15, 2015.

This work was supported by the National Natural Science Foundation of China (Grant No. 61271175), the Fundamental Research Funds for the Central Universities (Grant No. 7214505101), and Open Project Program of Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory (ITD-U14006/KX142600013).

Ma Zhuo (corresponding author, zma@mail.xidian.edu.cn) and Du Shuanyi (shydy@xidian.edu.cn) are with the State Key Lab of ISN, Xidian University, Xi'an and the Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, Shijiazhuang, China.

Many optimal or suboptimal reliability-based decoding algorithms are currently used for the concatenated decoding of LDPC codes, including the box-and-match algorithm (BMA) [9], chase-2 [10], and KNIH algorithms [11].

Most of the aforementioned cited studies evaluate the performance of a concatenated decoding algorithm by block error rate (BLER) or codeword error rate (CER). Given that the reliability-based decoding algorithms always return a valid codeword, once this codeword is incorrect (an undetected error occurs), more error bits will be introduced than that of a BP decoder. Therefore, the performance gain achieved in a bit error rate (BER) metric is not as striking as that in a BLER metric assuming the same decoding scheme.

References [12] and [13] consider bounded maximum-likelihood (ML) decoders, such as a bounded-distance ML decoder, a bounded-angle ML decoder, and a bounded reciprocal likelihood ratio ML decoder. All of these bounded ML decoders introduce a bound to detect whether a decoding codeword is correct. When a decoding metric exceeds a given bound, a code error is detected, and an erasure is claimed.

This paper introduces the idea of the bounded decoder to the OSD algorithm. A bound or threshold is used to determine whether to adopt the result of the OSD decoder or that of the BP decoder, based on serially concatenated BP-OSD (SC-BP-OSD) algorithms. Unlike in [12] and [13], the threshold in the algorithm proposed in this paper is dynamically calculated to avoid mismatching.

II. SC-BP-OSD Algorithm with Threshold

Given that OSD is serially concatenated with BP decoding, it is performed only when a BP decoding achieves its maximum iterations and the checksum of the related final output codeword is not zero. The following analysis concerns the case where OSD is evoked.

As previously discussed, OSD may choose a codeword other than the transmitted codeword as its decoding result, and the differences between these two codewords is no less than the minimum Hamming distance of the related LDPC code. Good LDPC codes often possess a large minimum Hamming distance. Thus, considerable bit errors occur once OSD returns an incorrect codeword. A predictable BER performance gain is obtained when an invalid codeword can be excluded and the BP decoding result is used. Finding a proper criterion for this choice is necessary.

An OSD algorithm contains the following two main steps. The first step is to construct the most reliable basis (MRB) and to then systematically reprocess candidate codewords expressed in the MRB [1]. In the second step, L_i candidate codewords are processed for order- i OSD of an LDPC code

with a code length of N , where

$$L_i = \sum_{l=0}^i \binom{N}{l}. \quad (1)$$

The OSD algorithm calculates a decoding metric for every candidate codeword and selects the codeword with the smallest decoding metric. For each constructed codeword, $\mathbf{c} = (c_0, c_1, \dots, c_{N-1})^T$, its associated decoding metric is computed based on the original noisy received sequence, $\mathbf{y} = (y_0, y_1, \dots, y_{N-1})^T$. For the SC-BP-OSD algorithm, \mathbf{y} is the accumulated LLR from BP decoding. The parameters \mathbf{r} and \mathbf{c}_1 are the absolute value and the hard decision of \mathbf{y} , respectively, which can be expressed as $r_i = |y_i|$ and $c_{1,i} = (1 - \text{sgn}(y_i))/2$, $0 \leq i \leq N-1$; the function $\text{sgn}(x)$ represents the sign of x . In fact, r_i is the reliability value of the corresponding $c_{1,i}$.

One of the most commonly used decoding metrics is that of the weighted Hamming distance, which can be defined as

$$d_w(\mathbf{c}_1, \mathbf{c}_x) \triangleq \mathbf{r}^T \cdot (\mathbf{c}_1 + \mathbf{c}_x), \quad (2)$$

where \mathbf{c}_x is one of the candidate codewords. The operation “+” is performed in GF(2). The decoding metric can be used to select the most likely codeword, which is denoted by \mathbf{c}_2 , among all of the L_i candidate codewords. Upon obtaining codeword \mathbf{c}_2 with the smallest decoding metric, this smallest decoding metric itself can also be an indicator for the reliability of \mathbf{c}_2 , which is a straightforward inference. Thus, a threshold, T_d , is set to filter the OSD result — if the final decoding metric (the decoding metric of \mathbf{c}_2), $d_w(\mathbf{c}_1, \mathbf{c}_2)$, is smaller than T_d , then the OSD result is chosen; otherwise, the BP decoding result is chosen.

Based on the previous analysis, an SC-BP-OSD algorithm with threshold is constructed. This algorithm can be expressed as SC-BP-OSD with threshold, and its flowchart is shown in Fig. 1.

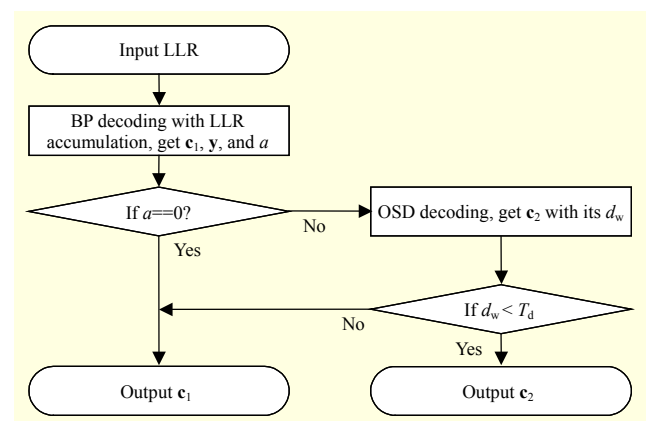


Fig. 1. Flowchart of SC-BP-OSD algorithm with threshold: variable “ a ” denotes the checksum of BP decoding result \mathbf{c}_1 .

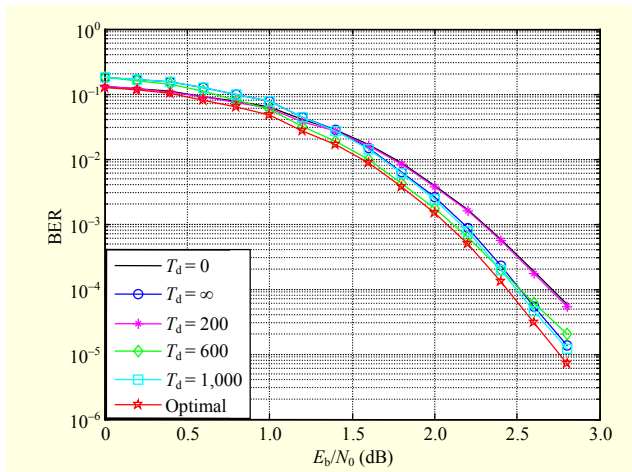


Fig. 2. Comparison of BER performance of different T_d .

The key issue of the SC-BP-OSD algorithm with threshold is determining a proper threshold T_d . A simple method is to investigate the performance of different T_d through simulation and select the best one. This work has been done for several famous LDPC codes. Unfortunately, the simulation results show that selecting a T_d with the best performance throughout the entire E_b/N_0 range is difficult.

Examples of utilizing decoding metrics to improve ML decoding algorithms are to be found in [14] and [15]. In [14], a threshold test is used to determine whether the codeword is likely to be the ML codeword. If the codeword is not accepted, then the decoder biases the input vector away from the candidate codeword, thereby reducing the decoding complexity. In [15], the OSD metric is used to arrange the list of the most a priori likely tests, thereby also reducing the decoding complexity. The algorithms in [14] and [15] are all based on pure OSD or ML decoders. Unlike in [14] and [15], the proposed approach of this paper uses the OSD metric to filter a correctly decoded codeword, which improves the BER performance of the SC-BP-OSD algorithm.

Figure 2 shows a comparison of the BER performance under an AWGN channel for different values of T_d , where an order-1 OSD is considered. The LDPC code used in Fig. 2 is a regular (3,6) LDPC code with a code rate of 1/2 and code length of 504 [16], represented as (504,252,3,6) LDPC code. The tested T_d in this paper includes 200, 600, and 1,000. In Figure 2, $T_d = 0$ represents a BP decoding without OSD, and $T_d = \infty$ represents an SC-BP-OSD algorithm without threshold. In an optimal situation, the decoding error caused by the OSD algorithm can always be detected, and this incorrect decoding codeword is then substituted with the BP decoding result. Although the *optimal* choice cannot be realized in practice, it can be computed in simulation. Whether the OSD output codeword is correct or not can be determined in simulation

because the simulator knows what is being transmitted. The result for the optimal situation can be a lower bound for evaluating the performance of different T_d .

Figure 2 shows that $T_d = 600$ performs better than $T_d = 200$ and $T_d = 1,000$ for E_b/N_0 less than 2.4 dB. However, for E_b/N_0 above 2.4 dB, $T_d = 1,000$ has the best performance. Thus, selecting the threshold is difficult. After all, the performance of both $T_d = 600$ and $T_d = 1,000$ has a notable gap from that of the optimal situation.

III. Dynamically Calculated Threshold

Since a fixed threshold cannot work properly in the entire E_b/N_0 region, a different T_d threshold can be used for different E_b/N_0 . However, this method requires many simulations to determine the T_d of different E_b/N_0 . Finally, using this method, the performance gap from the optimal algorithm cannot be reduced yet.

Based on the Gaussian approximation of the density evolution of the BP decoding algorithm of LDPC codes, and given an E_b/N_0 and an LDPC code, there will be a unique value for the mean of the LLR output, m_v , when the iteration approaches infinity [17]. The relationship between E_b/N_0 and m_v can be denoted as a function, $m_v = f(E_b/N_0)$. Based on the analysis in [17], the function $f(E_b/N_0)$ is monotonically increasing.

Considering that the Gaussian approximation of density evolution is derived based on the assumption that an all-zero codeword is transmitted, m_v is indeed a conditional mean. If a non-zero codeword is transmitted, then m_v can be approximated as the mean of the absolute values of the LLR outputs. As a result, the decoding metric, which is expressed by a weighted Hamming distance, also exhibits a growing trend because the weighting value is increased. This result explains why $T_d = 1,000$ performs better for large E_b/N_0 than $T_d = 600$.

The effect of the mean of the absolute values of the LLR outputs on the value of the decoding metric produces the idea that a variable threshold T_d based on the LLR output of BP decoding can be calculated as follows:

$$T_d = \alpha \cdot \frac{1}{n} \sum_{i=0}^{N-1} |r_i| = \alpha \bar{r}, \quad (3)$$

where \bar{r} is the mean of the absolute values of the LLR outputs (\mathbf{r}). The parameter α is to be determined and may be called the threshold factor.

The variable $d_w(\mathbf{c}_1, \mathbf{c}_2)$ can be expressed in the form of the sum of each element; that is,

$$d_w(\mathbf{c}_1, \mathbf{c}_2) = \sum_{i=0}^{N-1} |r_i| (c_{1,i} + c_{2,i}). \quad (4)$$

The following definition is then obtained:

$$d_{\text{cmp}} \triangleq \frac{1}{n} \sum_{i=0}^{N-1} |r_r| \cdot \sum_{i=0}^{N-1} (c_{1,i}, c_{2,i}) = d_h(\mathbf{c}_1, \mathbf{c}_2) \bar{r}, \quad (5)$$

where $d_h(\mathbf{c}_1, \mathbf{c}_2)$ is the Hamming distance between the codewords \mathbf{c}_1 and \mathbf{c}_2 . The variable d_{cmp} can be viewed as an estimation of $d_w(\mathbf{c}_1, \mathbf{c}_2)$.

A comparison of (3) and (5) shows that d_{cmp} and T_d are both scalar multiples of \bar{r} . Considering that the purpose of T_d is to determine whether the decoding metric d_w is too large to be accepted, the value of the scalar of T_d , α , has a close relationship with the scalar of d_{cmp} , $d_h(\mathbf{c}_1, \mathbf{c}_2)$.

If \mathbf{c}_1 falls inside the decision boundaries of the correct codeword \mathbf{c}_0 , then the Hamming distance between them is not more than $d_h(\mathbf{c}_1, \mathbf{c}_2)/2$. Under this assumption, \mathbf{c}_2 with decoding metric $d_w(\mathbf{c}_1, \mathbf{c}_2) > d_{\text{min}} \bar{r}/2$ has a large probability of being an incorrect codeword. Here, d_{min} is the minimum Hamming distance. Since a weighted Hamming distance is used in determining which codeword is more similar to \mathbf{c}_1 instead of a Hamming distance, a \mathbf{c}_1 with a Hamming distance from \mathbf{c}_0 of more than $d_{\text{min}}/2$ may also be correctly decoded. As a result, $\alpha > d_{\text{min}}/2$.

The lower bound of α has been retrieved, but how about the upper bound? Simulation and analysis do not suggest the existence of a proper upper bound that is suitable to all kinds of LDPC codes. However, the lower bound does facilitate the search for the best α because the simulated α can be adopted starting from the value that is half of the minimum Hamming distance of the considered LDPC code.

IV. Performance Analysis for Codeword Error Detection

The threshold of the algorithm proposed in this paper determines whether a decoding codeword is correct; however, this decision itself may or may not be a correct one. If an OSD output codeword is incorrect and is detected as an error through the threshold check, then a detected error occurs — the probability of which is expressed as P_d^+ . If a decoding codeword is incorrect but is not detected by a threshold, then an undetected error occurs; the probability of this happening is expressed by P_u . This definition of an undetected error is the same as that found in [12]–[13]; however, a detected error in [12]–[13] is slightly different in that there is no false detection probability, since detected errors coincide with erased codewords, and no decoder output is produced in such cases. Through the proposed approach, instead, a false detection is possible, since it may happen that an error is detected due to the decoding metric overcoming the threshold despite a decoder output codeword coinciding with a transmitted codeword.

Thus, a detected error rate, P_d , can be divided into two parts: one is P_d^+ , which is defined above, and the other is the probability that a decoding codeword is correct but is detected as an error, which can be denoted as P_d^- . The following relationships can now be established:

$$P_d = P_d^+ + P_d^-, \quad (6)$$

$$P_w = P_d + P_u = P_d^+ + P_d^- + P_u, \quad (7)$$

where P_w is the codeword error rate.

References [12]–[13] provide the relationships among P_w , P_d , and P_u for different bounds (which are of the same meaning as the threshold in this paper). The goal of [12]–[13] is to simultaneously minimize both P_w and P_u and to show the lower bound of P_u versus P_w for different bounds. Although different bounds are considered, the key point in [12]–[13] is not the bounds themselves but the relationship between P_w and P_u under such bounds. Based on the results of [12]–[13], a single bound will have a different performance under different E_b/N_0 . If a similar performance of P_u versus P_w for different E_b/N_0 is expected, then a bound should be varied with E_b/N_0 , which leads to a dynamic bound.

Unlike [12]–[13], OSD is serially concatenated with BP for the proposed approach; thus, an undetected codeword error should include that which is caused by BP. For a BP decoder, an undetected error occurs when $a = 0$; however, in such a case, the decoded codeword does not coincide with the transmitted one. For both the LDPC codes evaluated in this paper (also for most LDPC codes with sufficient code length), when only BP decoding is considered, the probability of an undetected codeword error occurring is small enough to be ignored. As a result, an undetected codeword error caused by BP is not considered in our later analysis and simulations.

For the proposed approach, P_u is expected to be as small as possible; that is, most of the decoding errors of OSD can be correctly detected. Moreover, P_d^- should be minimized, which means the minimizing of the probability of discarding the correct OSD output codeword. The so-called optimal situation implies that P_u and P_d^- are all zero and that all the codeword errors are correctly detected errors. Unfortunately, the decline of P_u results in an increase in P_d^- , which in turn, is responsible for the increase in P_w shown in [12]. If the CER of OSD without threshold is P_{w_osd} , then

$$P_{w_osd} = P_d^+ + P_u. \quad (8)$$

Considering that P_{w_osd} is determinate given an E_b/N_0 , the proportion of P_u , P_d^- , and P_d^+ in P_w is clear enough to indicate the influence of the threshold on the proposed approach. Let P_{c_u} , $P_{c_d^-}$, and $P_{c_d^+}$ be a percentage of P_u , P_d^- , and P_d^+ , respectively, in P_w ; Figure 3 shows them for different thresholds.

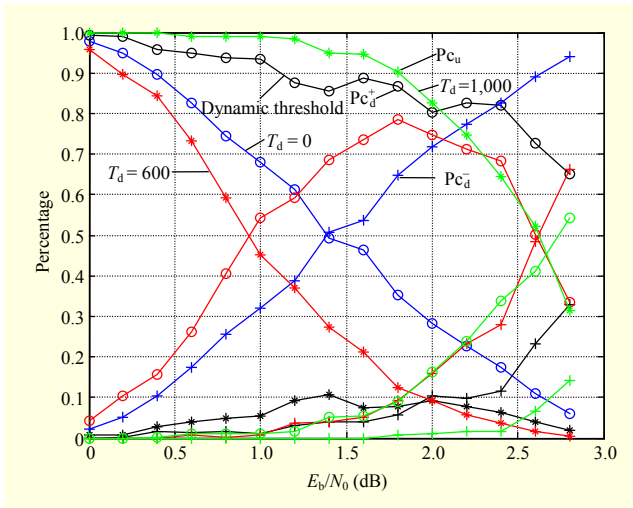


Fig. 3. Comparison of P_{c_u} , $P_{c_d^-}$, and $P_{c_d^+}$ for different thresholds. Black line denotes result of dynamic threshold; blue line that of $T_d = 0$; red line that of $T_d = 600$; and green line that of $T_d = 1,000$. For every situation above, the lines with *, +, and o denote the percentages of P_u , P_{d^-} , and P_{d^+} , respectively.

The (504,252,3,6) LDPC code used in Fig. 2 is also selected for computing P_{c_u} , $P_{c_d^-}$, and $P_{c_d^+}$ in Fig. 3, and order-1 OSD is concerned. The P_{c_u} , $P_{c_d^-}$, and $P_{c_d^+}$ for the optimal situation and for $T_d = \infty$ are not shown in Fig. 3, because the following results are easy to be obtained:

$$\begin{cases} P_{c_u} = 0, P_{c_d^-} = 0, P_{c_d^+} = 1 & \text{optimal,} \\ P_{c_u} = 1, P_{c_d^-} = 0, P_{c_d^+} = 0 & T_d = \infty. \end{cases} \quad (9)$$

Percentage P_{c_u} for $T_d = 0$ is always zero; thus, it is also not shown in Fig. 3.

Figure 3 shows that the $P_{c_d^+}$ for the dynamic threshold situation is above 80% for E_b/N_0 less than 2.4 dB and significantly decreases when E_b/N_0 reaches 2.4 dB. This decreasing is caused by the increase in $P_{c_d^-}$. However, the performance of $P_{c_d^+}$ in the dynamic threshold situation is the best among all of the three given situations, which means that the filtering of the OSD output codeword has been optimized. For constant thresholds, such as $T_d = 600$ and $T_d = 1,000$, the fluctuation of $P_{c_d^+}$ with E_b/N_0 is on a large scale. As a result, $T_d = 600$ and $T_d = 1,000$ are only suitable for a part of the entire E_b/N_0 region.

The P_{c_u} performance for the dynamic threshold situation is far better than that for the two constant thresholds situations, which is less than 11% for the whole E_b/N_0 region.

In all the situations, the variation of $P_{c_d^-}$ with E_b/N_0 exhibits a similar trend; that is, $P_{c_d^-}$ increases as E_b/N_0 grows. This phenomenon implies that, for large E_b/N_0 , correctly detecting the decoding codeword error is difficult. Fortunately, the decoding bit error of BP itself for large E_b/N_0 is also small,

which alleviates the bit error caused by P_{d^-} .

The situation with $T_d = 0$ has a constant P_{c_u} , which is always zero. This makes $P_{c_d^-}$ and $P_{c_d^+}$ complementary. The two curves of $P_{c_d^-}$ and $P_{c_d^+}$ constitute an “X” form in the figure, with its crossing point near $E_b/N_0 = 1.3$ dB. This “X” form indicates that the OSD output codeword becomes increasingly more accurate as E_b/N_0 increases.

V. Simulation Results

Two rate 1/2 LDPC codes are simulated to verify the effectiveness of the proposed approach. Both of them have a code length of 504. Code A is the (504,252,3,6) LDPC code used in Fig. 2. Code B is an irregular LDPC code constructed using the progressive edge-growth algorithm [16]. The simulation is performed under an AWGN channel. The

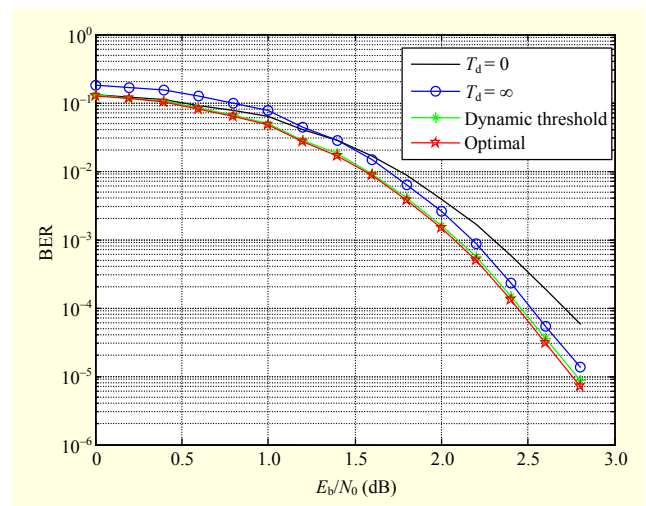


Fig. 4. BER performance of code A for different thresholds.

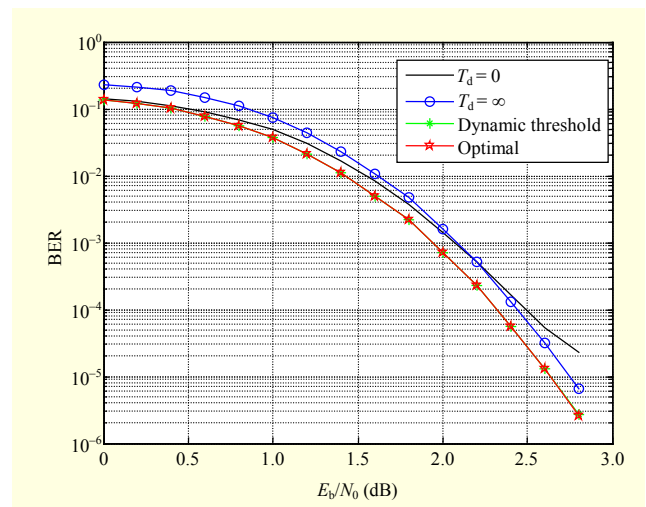


Fig. 5. BER performance of code B for different thresholds.

simulation result is shown in Figs. 4 and 5. The threshold factors (α) used for computing the dynamic threshold in Figs. 4 and 5 are 16 and 30, respectively.

Figures 4 and 5 show that the BER performance of the SC-BP-OSD algorithm can be improved by introducing a dynamically calculated threshold for both codes. The BER improvement is approximately 0.1 dB for code A and approximately 0.15 dB for code B. For both codes, the BER performance of the proposed algorithm can approach the performance of the optimal situation. Particularly for code B, the BER curve of the dynamic threshold situation almost coincides with that of the optimal situation.

Figures 4 and 5 also show that the BER performance of the SC-BP-OSD algorithm without threshold is worse than that of the pure BP algorithm for small E_b/N_0 . This result is different from the BLER performance of the SC-BP-OSD algorithm, where the SC-BP-OSD always performs better than BP. However, with a dynamically calculated threshold, the approach proposed in this paper can also outperform BP for all E_b/N_0 regions when BER is considered.

VI. Conclusion

A dynamically calculated threshold is introduced in this paper to improve the BER performance of a serially concatenated BP-OSD algorithm. Most of the current improvements for the concatenated decoding algorithm are considered in relation to BLER performance, not BER. When BER is considered, most of the existing algorithms become mediocre in light of what has been achieved under the BLER metric. Using a threshold to determine whether the OSD output codeword is credible is a straightforward idea, but the simulation results show that a constant threshold cannot perform well for all E_b/N_0 regions. Therefore, a dynamic threshold is computed from the mean of the absolute values of the LLR outputs of BP decoding. The codeword error detection analysis shows that SC-BP-OSD with dynamic threshold has better $P_{c_d}^+$ performance, which in turn, leads to better BER performance than that with constant threshold. Using the dynamic threshold, the BER performance of two rate 1/2 LDPC codes can approach that under the optimal situation.

In addition, the computation of the dynamic threshold requires only the mean of the absolute values of the input LLR of OSD. As a result, the increase in the computational complexity of the proposed approach is negligible compared with the high complexity of the OSD algorithm itself.

References

[1] M. Fossorier, "Iterative Reliability-Based Decoding of Low-

- Density Parity Check Codes," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, May 2001, pp. 908–917.
- [2] T.J. Richardson and R.L. Urbanke, "The Capacity of Low-Density Parity-Check Codes under Message-Passing Decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, Feb. 2001, pp. 599–618.
- [3] S. Gounai and T. Ohtsuki, "Lowering Error Floor of Irregular LDPC Codes by CRC and OSD Algorithm," *IEICE Trans. Commun.*, vol. E89B, no. 1, Jan. 2006, pp. 1–10.
- [4] S. Gounai and T. Ohtsuki, "Decoding Algorithms Based on Oscillation for Low-Density Parity Check Codes," *IEICE Trans. Commun.*, vol. E88A, no. 8, Aug. 2005, pp. 2216–2226.
- [5] M. Jiang et al., "Reliability-Based Iterative Decoding of LDPC Codes Using Likelihood Accumulation," *IEEE Commun. Lett.*, vol. 11, no. 8, Aug. 2007, pp. 677–679.
- [6] G.W. Li and G.Z. Feng, "Generalised Reliability-Based Syndrome Decoding of LDPC Codes," *European Trans. Telecommun.*, vol. 19, no. 8, Dec. 2008, pp. 873–877.
- [7] L. Yang, M. Tomlinson, and M. Ambroze, "Decoding Low-Density Parity-Check Codes with Error-Floor Free over the AWGN Channel," *Proc. WCNIS*, Beijing, China, June 25–27, 2010, pp. 192–196.
- [8] M. Baldi et al., "A Hybrid Decoding Scheme for Short Non-binary LDPC Codes," *IEEE Commun. Lett.*, vol. 18, no. 12, Dec. 2014, pp. 2093–2096.
- [9] Y.B. Bian and G.Z. Feng, "A Novel Concatenation Decoding Algorithm for Short LDPC Codes with Lower Complexities," *Proc. PCCF*, Nanning, China, Sept. 2009, pp. 550–554.
- [10] S. Tong and H.J. Zheng, "On Combining Chase-2 and Sum-Product Algorithms for LDPC Codes," *ETRI J.*, vol. 34, no. 4, Aug. 2012, pp. 629–632.
- [11] G.L. Qiao, "Parallel Decoding Scheme Based on OSD and KNIH Algorithms," *Adv. Mater. Res.*, vol. 433–440, Jan. 2012, pp. 4813–4816.
- [12] S. Dolinar et al., "The Limits of Coding with Joint Constraints on Detected and Undetected Error Rates," *Proc. ISIT*, Toronto, Canada, July 6–11, 2008, pp. 970–974.
- [13] S. Dolinar et al., "Bounds on Error Probability of Block Codes with Bounded-Angle Maximum-Likelihood Incomplete Decoding," *Proc. ISITA*, Auckland, CA, USA, Dec. 7–10, 2008, pp. 1–6.
- [14] R. Kerr and J. Lodge, "Near ML Performance for Linear Block Codes Using an Iterative Vector SISO Decoder," *Proc. Turbo Coding*, Munich, Germany, Apr. 3–7, 2006, pp. 1–6.
- [15] A. Kabat, F. Guilloud, and R. Pyndiah, "New Approach to Order Statistics Decoding of Long Linear Block Codes," *Proc. IEEE GLOBECOM*, Washington, DC, USA, Nov. 26–30, 2007, pp. 1467–1471.
- [16] S.-Y. Chung, T.J. Richardson, and R.L. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using Gaussian Approximations," *IEEE Trans. Inf. Theory*, vol.

47, no. 2, Feb. 2001, pp. 657–670.

- [17] D.J.C. MacKay, Encyclopedia of Sparse Graph Codes. Accessed May 11, 2013. <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>



Ma Zhuo received his BS degree in communication engineering and his MS and PhD degrees in communication and information systems from Xidian University, China, in 2003, 2006, and 2012, respectively. He joined the State Key Laboratory of ISN, Xidian University, as an assistant in 2006 and became a lecture in 2009. His research interests include channel coding and signal processing in wireless communication.



Du Shuanyi received his BS degree in communication engineering and his MS degree in communication and electronic systems from Xidian University, China, in 1983 and 1990, respectively. He joined the School of Communication Engineering, Xidian University, in 1983. He is now working as a professor at the State Key Laboratory of ISN, Xidian University. His research interests include wireless digital communication and digital signal processing.